

IN THE CLAIMS:

1. (Currently Amended) A system comprising:

A first preferred memory of a given size and a second memory;

a data filter coupled to a text-to-image converter for converting filtered data into image data using texture maps; and

a processor coupled to the data filter, wherein the processor executes a first logic in which the total size of the set of texture maps is less than or equal to the first memory size;
and

a second logic if the total size of the set of texture maps is greater than the first memory size, to divide the set of texture maps into two groups and

a memory analyzer for analyzing set-top box layout to determine the types and corresponding sizes of ~~memory~~ at least the first and the second memories available in the set-top box, the memory analyzer being coupled to a memory distributor, wherein the system determines a storage size necessary for storing the texture maps defining one or more images and the memory distributor distributes the texture maps to a preferred first memory if the total size of the texture maps is less than or equal to the available first memory size and further wherein if the total size of the texture maps is greater than the first memory size, the texture maps ~~being~~ are divided into two groups where a first group's total size is the largest possible sum of map sizes less than the first memory size and where a second group's total size is the difference between the total size of the set of texture maps and the total size of the first ~~group~~ group, wherein the set of texture maps of the first group is stored in the first memory, wherein the set

of texture maps of the second group is stored in the second memory, and wherein the set of texture maps of the second group is compressed to fit into the first memory if sufficient second memory is not available to receive the set of texture maps of the second group.

2. (Original) The system of claim 1, wherein a total size of the set of texture maps is a sum of all texture map sizes.

Claims 3-8 (Canceled)

9. (Original) The system of claim 7, further comprising a compression engine.

10. (Currently Amended) A method comprising:

computing a total size of a set of texture maps;

searching for types of memories in a set-top box;

comparing the total size of the set of texture maps with a size of a preferred first

memory;

dividing the set of texture maps into at least two groups if the total size of the set of texture maps is larger than the first memory size, such that the total size of the texture maps in a first group is the largest possible sum of texture map sizes for which the total size of texture maps is less than the first memory size and the total size of a second group is the difference between the total size of the set of texture maps and the total size of the first ~~group~~ group,

storing the second group of texture maps in a second memory, and compressing the second

group of texture maps to fit into the first memory if sufficient second memory is not available.

11. (Original) The method of claim 10 wherein computing a total size of a set of texture maps comprises:

computing a sum of all texture maps.

12. (Previously Presented) The method of claim 10 further comprising:

storing the set of texture maps in the first memory if the total size of the set of texture maps is less than or equal to the first memory size.

13. (Previously Presented) The method of claim 10 further comprising:

storing the first group of texture maps in the first memory.

14. (Canceled)

15. (Canceled)